



LATVIJAS
UNIVERSITĀTE

Zemkvadrātisks algoritms koku ceļu apakšvirkņu uzdevumam

Autors: Aleksejs Zajakins
Studenta apl. nr.: az11182

Darba vadītājs: LU docents,
Dr. dat. Jevgēnijs Vihrovs

Ievads

Kursa darbā tiek aplūkoti uzdevumi, kas ir saistīti ar viena koka iekļaušanu citā kā apakškoku vai grafa minoru. Šādi uzdevumi bieži parādās dažādos kontekstos, ieskaitot molekulāro bioloģiju, ķīmiju un XML dokumentu apstrādi; tāpēc arī algoritmiem, kas risina šādus uzdevumus, ir iespējami dažādi praktiski pielietojumi.

Divu koku lielākā kopīgā apakškoka atrašanas uzdevumu var uzskatīt par lielākās kopīgās apakšvirknes atrašanas uzdevuma (LCS — longest common subsequence) vispārinājumu. Vispārīgā gadījumā LCS ir risināms $O(n^2)$ laikā, tāpēc arī lielāko kopīgo apakškoku daudzus gadījumos nevar atrast ātrāk par $O(n^2)$ laiku. Savukārt, gadījumā ja vismaz viena no ieejas virknēm ir permutācija, LCS uzdevums ir ekvivalents ar lielākās augošas apakšvirknes atrašanas uzdevumu (LIS — longest increasing subsequence), kas jau ir risināms $O(n \log n)$ laikā. Līdzīgi, arī lielāko kopīgo apakškoku ir iespējams atrast ātrāk dažādos speciālgadījumos.

Kursa darbā tiek izpētīta literatūra un tiek aprakstīti kādi precīzi uzdevumi tiek risināti eksistējošos rakstos un kādu sarežģītību algoritmi tajos ir aprakstīti.

Aplūkoti uzdevumi

- **Largest Common Subtree** — atrast lielākā izmēra mežu, ko var iegūt no abiem ieejas kokiem izdzēšot virsotnes. Eksistē polinomiāli algoritmi (vismaz kvadrātiski), kas risina uzdevumu saknotu sakārtotu koku gadījumā.
- **Largest Agreement Subtree** — kokiem, kuriem ir marķētas tikai lapas, atrast apakškoku ar lielāko lapu skaitu, ko var iegūt no abiem ieejas kokiem izdzēšot lapas un aizstājot iekšējās virsotnes ar pakāpi divi ar šķautnēm. Eksistē polinomiāli algoritmi (vismaz kvadrātiski) patvaļīgiem kokiem un $O(n \log n)$ algoritms bināriem kokiem.
- **Maximum Common Embedded Subtree** — atrast lielāko apakškoku, kas ir iekļauts abos ieejas kokos kā grafa minors. Saknotu sakārtotu koku gadījumā eksistē $O(n^4)$ algoritms.
- **Tree Path Subsequence Problem** — noteikt kuri pirmā koka ceļi ir apakšvirknes kuriem otrā koka ceļiem. Tiek apskatīti tikai saknoti koki un tikai tādi ceļi, kas sākas saknē un beidzas lapā. Eksistē algoritmi ar sliktākā gadījuma sarežģītību $O(n^2 / \log n)$.

Jauns uzdevuma speciālgadījums

- **Largest Common Embedded Path** — atrast garāko ceļu, kas ir apakšvirke kādam ceļam katrā no ieejas kokiem. Šāds uzdevuma speciālgadījums iepriekš literatūrā netika apskatīts. Darba autors ir atradis algoritmu, kas risina šo uzdevumu laikā $O(n \log^3 n)$.

